Airbnb - Edinburgh

Attila Szuts

31/01/2021

Contents

Intro

In this assignment I am going to investigate Airbnb listings from Edinburgh on 27 January, 2020. I chose this city as I have long wanted to visit it, and hopefully after the pandemic I will be able to travel there and perhaps this short analysis could be useful in finding good deals among aibnb listings. I had to use an earlier date for my model as there were quite few listings recently, no wonder. The task was to build a model that can price apartments between 2-6 guests in a given city.

I am going to build three different models and compare their performance in terms of RMSE and pick the one with the best predictive power on a test set. The first model will be a multiple regression model with hand picked coefficients. The second model will be a multiple regression model using LASSO and finally the third model will be built using random forests approach.

Data prep

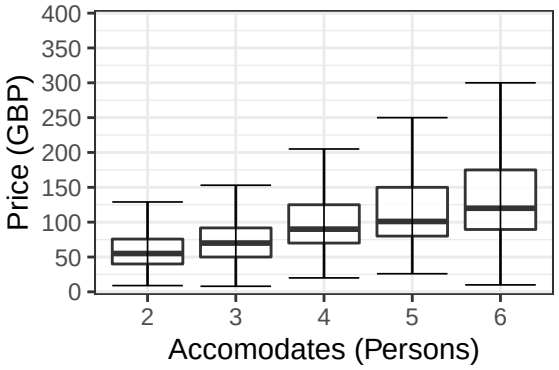
The total number of observations before exclusion were 13208. Data cleaning and feature engineering consisted of the following steps/decision points:

1. Remove $-sign from price related variables (price, cleaning fee, extra people)
2. Property type filtered for only Apartment.
3. Room type converted to factor and factor levels shortened.
4. Cancellation policy converted to factor and super strict category created from `super_strict_30` and `super_strict_60`
5. Bed type and neighbourhood converted to factors.
6. Created variable number of days since first review.
7. Create dummy variables from amenities. Kept only those that were in at least 80 listings. I came up with this number from the histogram of total dataset so that I keep the 100 most frequent amenities.
8. I filtered aparments to only include those that are between 2 and 6 persons.
9. Filtered property types to only include Apartments.
10. Filtered out expensive listings (price > 500 GBP).

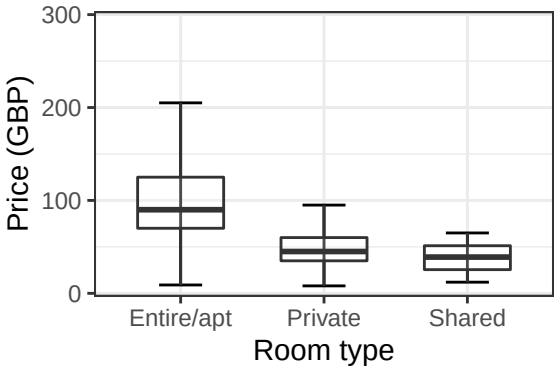
All code is available on github. The total number of observations after filtering was 12221.

Descriptives

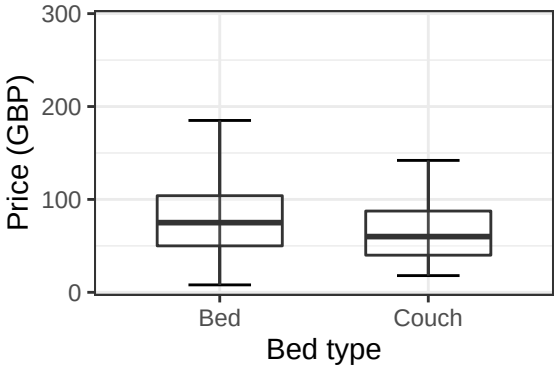
Price increases as the number of people accomodated increases, altough the spread increases too.



There is not much difference in price between Private and shared rooms, however Entire homes are more expensive than both, again with a much higher variance, too.



Interestingly, there is not much difference in price between apartments that have a Bed or just a Couch. Note, that there are very few observations without beds in this dataset.



Regression

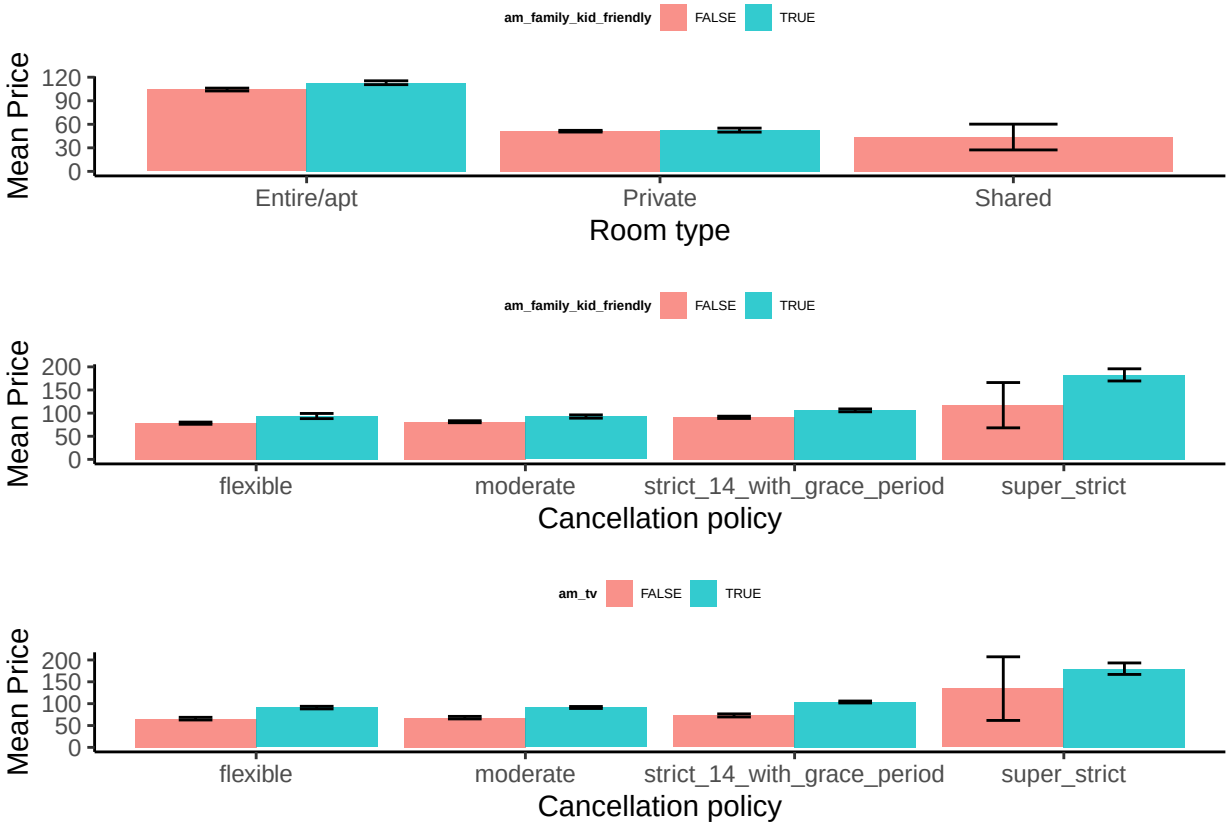
In the first model I am going to include coefficients based on the case study that Gabor Bekes and Gabor Kezdi has done. However, I am going to look for interactions in this dataset, instead of using theirs.

As we can see, the interactions used by Gabors are not prevalent in this dataset. This is an interesting external validity finding regarding their models. Even though those were very obvious interaction terms in London, they are not very relevant in Edinborough. However, I am still going to include them in my regression model that is going to use LASSO to find the best possible model to see if maybe there are other interactions that could be useful.

I built three separate models:

1. a baseline model, only using the number of people that the apartment can accommodate as a predictor
2. an extended model with all basic data and amenities.
3. and a complete model with all of the second model's predictors + the interactions.

I created a holdout dataset with 20% of observations to evaluate the models on.



Using 5-fold cross validation I checked which one of my three regression models performed best.

In the table below we can see the three models I have built. We can see that:

1. The second model performs better than the first in terms of variance explained and also in terms of test RMSE. That is, the second model is overall more useful than the first.
2. The second model performs worse than the third in terms of variance explained, however it is much better in terms of test RMSE. Put another way, the third model is overfitting the data.

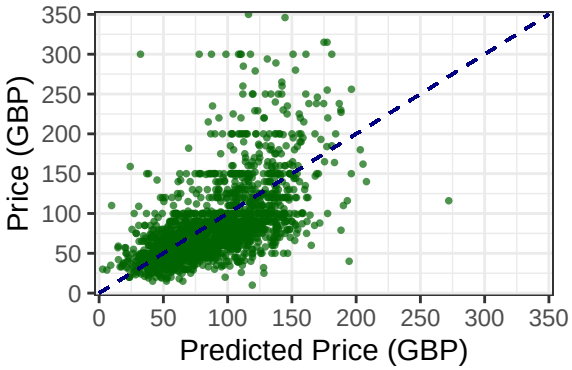
	Model	N predictors	R-squared	BIC	Training RMSE	Test RMSE
1	(1)	1	0.22	80038	51.98	51.91
2	(2)	105	0.35	71494	45.77	46.59
3	(3)	489	0.40	74343	43.77	66.61

Regression with LASSO

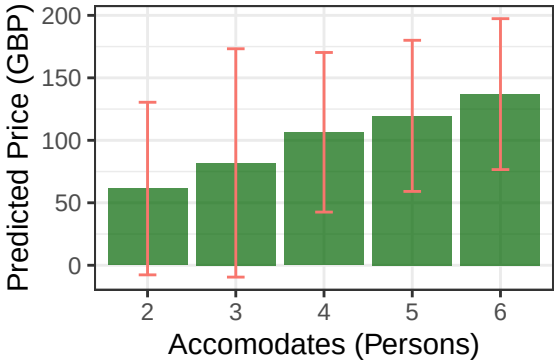
When doing regression with LASSO, I started out with the second model, and added all interaction terms to the model.

Model Diagnostics

Taking a look at Diagnostic data, we can see that our simpler model performed better (45.4601132) in terms of RMSE than LASSO (66.2614122). Our model performed similarly to the model presented in the London Case study, that is, it predicts apartments' prices remarkably well - if they are relatively cheap. It performs quite poorly, however, as price increases. This is most likely due to the fact, that there are quite few expensive apartments in the dataset.



Predicted prices for different sized apartments vary in terms of prediction accuracy. Smaller apartments are priced much worse in comparison to bigger appartments. This is in part because of the linear nature of the model: the same sized prediction interval can mean negative values for cheaper (smaller) appartments. This is an expected outcome with linear models, and there is not much you can do. However, here, the size of the PI is different depending on the number of people accommodated by the appartment.



Accomodates	Prediction	Pred. interval lower	Pred. interval upper	Conf.interval lower	Conf.interval upper
2	61.390	-7.675	130.455	38.389	84.391
3	81.895	-9.477	173.267	36.704	127.085
4	106.436	42.570	170.302	89.790	123.083
5	119.573	59.095	180.050	106.288	132.858
6	136.975	76.586	197.364	124.244	149.706

Random forest

As the third model that I am going to build now, I will create a Random Forest using caret. I will use the same dataset to see if there is a meaningful increase in performance to justify the cost of increased running time compared to OLS.

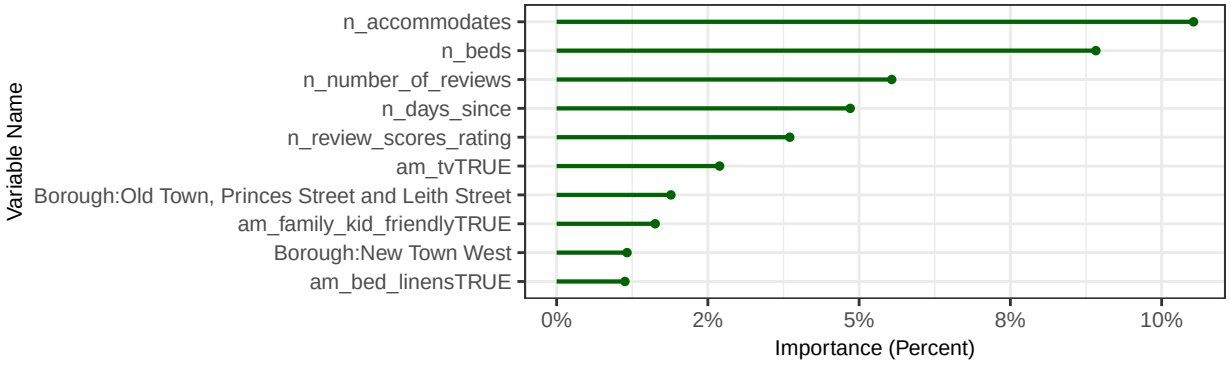
I created three different RF models as well, to choose the best that I can compare with regression models. The first one only incorporated basic variables, the second had the same variables as the second linear model, and the third was the same as the second but with autotuning.

In the table below we can see the results of the three different RF model that I've built. Based on RMSE alone, the third model (autotuned) performed the best, however not significantly better.

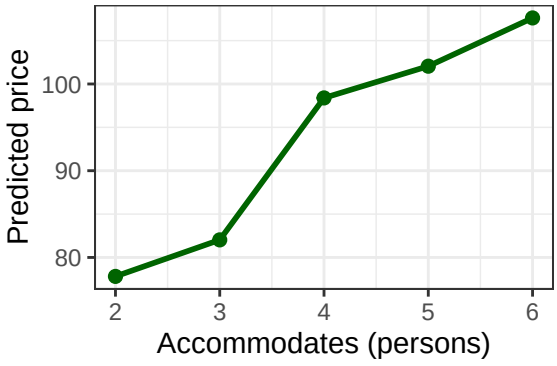
	RMSE
Model A	48.241
Model B	45.912
Model B auto	44.857

Model Diagnostics

On the variable importance plot we can see what are the most frequently used variables to split the trees. The number of people accomodated is the most frequently used, after that the number of beds and then the number of reviews. This gives us an idea what are the most important 'coefficients' in the model.



The partial dependence plots shows the average predicted price for different sized appartments. We can use this as a starting point when trying to find an answer to the question of pricing appartmetns.



Compare models

And finally we can see the results of the comparisons in the table below. The best model seems to be the simplest Random Forest, based on the holdout RMSE. It is interesting, however, that it performs the worst in the Cross Validation. This suggests that this is the most robust model, because even though it is worse on CV, it's the best in a 'real-world' scenarios, consequently I am going to pick this model as a winner.

It is worth mentioning the case of external validity. There are two questions that should be answered. Are these findings generalizable over time and over space? The first question could be easily answered: just download a different date from this city, run the models and evaluate the results. The same thing could be done for a different city, but I think there would be substantial differences in the model. For example, one important predictor is the neighbourhood, which is different in each city. So it would be interesting to investigate an integration of models to be able to generalize findings across different cities.

Note: I wasn't able to figure out the reason why the table prints out a different Holdout RMSE, it should be: 66.261.

	CV RMSE	Holdout RMSE
OLS	46.544	66.483
LASSO (model w/ interactions)	46.315	26074.925
Random forest (smaller model)	48.691	62.644
Random forest	46.068	63.049
Random forest (auto tuned)	46.080	65.942